

2  
DIALOG(R)File 351:Derwent WPI  
(c) 2001 Derwent Info Ltd. All rts. reserv.

011792343 \*\*Image available\*\*  
WPI Acc No: 1998-209253/199819  
XRPX Acc No: N98-166304

Resource management and recovery method for distributed systems -  
involves requesting agreement resource lease periods with manager and  
having resource reclaimed when all leases have expired

Patent Assignee: SUN MICROSYSTEMS INC (SUNM )

Inventor: RIGGS R; WALDO J H; WOLLRATH A M

Number of Countries: 025 Number of Patents: 004

Patent Family:

| Patent No          | Kind | Date        | Applicat No | Kind     | Date     | Week     |
|--------------------|------|-------------|-------------|----------|----------|----------|
| EP 836140          | A2   | 19980415    | EP 97402364 | A        | 19971008 | 199819 B |
| US 5832529         | A    | 19981103    | US 96729421 | A        | 19961011 | 199851   |
| <b>JP 11045187</b> | A    | 19990216    | JP 97293546 | A        | 19971010 | 199917   |
| US 6081813         | A    | 20000627    | US 96729421 | A        | 19961011 | 200036   |
|                    |      | US 98152062 | A           | 19980911 |          |          |

Priority Applications (No Type Date): US 96729421 A 19961011; US 98152062 A  
19980911

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes

EP 836140 A2 E 20 G06F-009/46

Designated States (Regional): AL AT BE CH DE DK ES FI FR GB GR IE IT LI

LT LU LV MC NL PT RO SE SI

US 5832529 A G06F-017/30

JP 11045187 A 18 G06F-009/46

US 6081813 A G06F-017/30 Cont of application US 96729421

Cont of patent US 5832529

Abstract (Basic): EP 836140 A

The distributed computer system has resources, e.g. files or memory, that are accessed by a number of computers. The system has to recover these items when they are no longer used. When a computer requires to use a resource, it sends a "dirty call" to the resource manager. This call increments a count on the resource. As part of the call the requester and manager agree a period for which the resource will be "leased", e.g. retained available to the requester.

The requester and manager monitor the lease period and the requester may send lease extending calls. When the lease expires the manager reclaims the resource.

ADVANTAGE - Provides reference integrity for accesses to the resource while ensuring that it can be recovered.

Dwg. 1/8

Title Terms: RESOURCE; MANAGEMENT; RECOVER; METHOD; DISTRIBUTE; SYSTEM;  
REQUEST; AGREE; RESOURCE; LEASE; PERIOD; MANAGE; RESOURCE; RECLAIM; LEASE  
; EXPIRE

Derwent Class: T01

International Patent Class (Main): G06F-009/46; G06F-017/30

International Patent Class (Additional): G06F-012/00; G06F-012/02

File Segment: EPI

Manual Codes (EPI/S-X): T01-F02C1; T01-H07C3; T01-M02A; T01-M02A1

10.

**THIS PAGE BLANK (USPTO)**

|               |       |              |
|---------------|-------|--------------|
| (51) Int.Cl.* | 識別記号  | F I          |
| G 0 6 F 9/46  | 3 6 0 | G 0 6 F 9/46 |
| 12/00         | 5 9 1 | 12/00        |
|               |       | 3 6 0 C      |
|               |       | 5 9 1        |

審査請求 未請求 請求項の数42 F D (全 18 頁)

(21) 出願番号 特願平9-293546

(22) 出願日 平成9年(1997)10月10日

(31) 優先権主張番号 08/729, 421

(32) 優先日 1996年10月11日

(33) 優先権主張国 米国 (U S)

(71) 出願人 597136607

サンマイクロシステムズ インコーポレー  
テッドアメリカ合衆国、94043 カリフォルニア  
州、マウンテン ビュー、エムエス ビー  
エーエルアイ-521、ガルシア アヴェニ  
ュー 2550

(72) 発明者 ウールラス アン エム

アメリカ合衆国、01450 マサチューセッ  
ツ州、グラットン、ノースウッズロード9

(74) 代理人 弁理士 上野 登

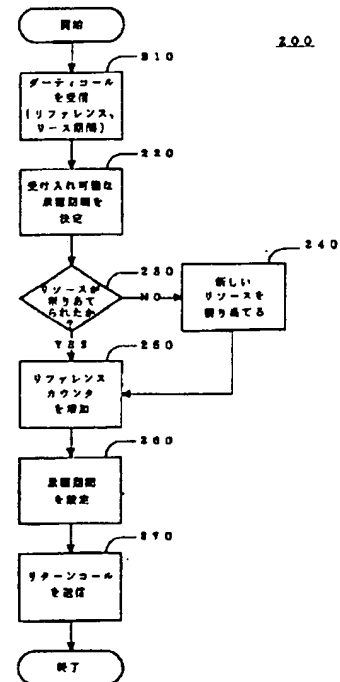
最終頁に続く

(54) 【発明の名称】 分散ガーベッジコレクションのためのリソースの管理方法、装置、コンピュータプログラム生産物、コンピュータシステム並びにプラットフォーム間に分散したリソースの割当及び割当解除方

(57) 【要約】

【課題】 記憶条件の追加や不必要な通信の増大を招くことなく、リファレンスの保安全性が維持され、かつ記憶漏れを起こすことのない分散ガーベッジコレクションのためのリソースの管理方法、装置、コンピュータプログラム生産物、コンピュータシステム並びにリソースの割当及び割当解除方法を提供すること。

【解決手段】 あるシステムのリソースを参照し、かつ要求されたリース期間を指定するリクエスタープラットフォームからの要求を受信する工程 (ステップ210) と、リース期間中はそのシステムリソースへの共有アクセスを許可する工程 (ステップ250、260) と、リクエスタープラットフォームに対してリース期間を通知するリターンコールを送信する工程 (ステップ270) と、リース期間が終了した時に、そのシステムリソースの割当を解除する工程とを備えるようにした。



## 【特許請求の範囲】

【請求項1】 リソースを参照し、かつ要求されたリース期間を指定するプロセスからの要求を受信する受信工程と、

承認されたリース期間、前記リソースに対する共有アクセスを許可する許可工程と、

前記プロセスに前記承認されたリース期間を通知する通知工程と、

前記承認されたリース期間が終了した時に前記リソースの割当を解除する割当解除工程とを備えていることを特徴とする分散ガーベッジコレクションのためのリソースの管理方法。

【請求項2】 前記許可工程は、前記要求されたリース期間よりも短い期間となるように前記承認されたリース期間を決定するサブ工程を備えていることを特徴とする請求項1に記載される分散ガーベッジコレクションのためのリソースの管理方法。

【請求項3】 前記許可工程は、前記要求されたリース期間、前記リソースの大きさ、及び前記リソースのための他の承認されたリース期間の少なくとも一つに基づき、前記承認されたリース期間を決定するサブ工程を備えていることを特徴とする請求項1に記載される分散ガーベッジコレクションのためのリソースの管理方法。

【請求項4】 前記許可工程は、前記リソースがそれ以前に割り当てられなかったという決定に基づいて前記リソースを割り当てるサブ工程を備えていることを特徴とする請求項1に記載される分散ガーベッジコレクションのためのリソースの管理方法。

【請求項5】 前記許可工程は、前記リソースに対応するリファレンスカウントを増加させるサブ工程を備え、前記リファレンスカウントは、現在、どれだけ多くのプロセスが前記リソースに対するリファレンスを持っているかを示すものであることを特徴とする請求項1に記載される分散ガーベッジコレクションのためのリソースの管理方法。

【請求項6】 前記プロセスがある識別されたリソースに対するアクセスの終了を望んでいることを示す前記プロセスからの第2の要求を受信する工程と、

前記識別されたリソースに対応する前記リファレンスカウントを減少させる工程とをさらに備えたことを特徴とする請求項5に記載される分散ガーベッジコレクションのためのリソースの管理方法。

【請求項7】 前記割当解除工程は、前記リソースに対する共有アクセスがもはや要求されない時に、前記リソースをリクレーンするためのガーベッジコレクションプロセスを呼び出すサブ工程を備えていることを特徴とする請求項1に記載される分散ガーベッジコレクションのためのリソースの管理方法。

【請求項8】 前記割当解除工程は、前記リファレンスカウントが前記識別されたリソースに、現在、いかなる

プロセスもアクセスしていないことを示している時に、前記識別されたリソースの割当を解除するためのガーベッジコレクションプロセスを呼び出すサブ工程を備えていることを特徴とする請求項6に記載される分散ガーベッジコレクションのためのリソースの管理方法。

【請求項9】 あるリース期間、あるリソースに対してアクセスすることをプロセスから要求する工程と、

前記リソースに対する共有アクセスが許可されている間、承認されたリース期間を前記プロセスから受信する工程と前記承認されたリース期間が終わりかけているが、前記リソースへのアクセスが完了していないという決定に基づいて、新たなリース期間の要求を前記プロセスに対して送信する工程とを備えていることを特徴とする分散ガーベッジコレクションのためのリソースの管理方法。

【請求項10】 前記承認されたリース期間が終了する前に、前記リソースに対する共有アクセスがもはや要求されないことを示す異なる要求を送信する工程をさらに備えていることを特徴とする請求項9に記載される分散ガーベッジコレクションのためのリソースの管理方法

【請求項11】 リソースを参照し、かつ要求されたリース期間を指定するプロセスからの要求を受信するように構成された受信モジュールと、

承認されたリース期間、前記リソースへの共有アクセスを許可するように構成されたリソースアロケータと、前記プロセスに前記承認されたリース期間を通知するように構成された通知モジュールと、前記承認されたリース期間が終了した時に前記リソースの割当を解除するように構成されたリソースディアロケータとを備えていることを特徴とする分散ガーベッジコレクションのための装置。

【請求項12】 前記リソースアロケータは、前記要求されたリース期間よりも短い期間となるように前記承認されたリース期間を決定するように構成された決定モジュールを備えていることを特徴とする請求項11に記載される分散ガーベッジコレクションのための装置。

【請求項13】 前記リソースアロケータは、前記要求されたリース期間、前記リソースの大きさ、及び前記リソースのための他の承認されたリース期間の少なくとも一つに基づき、前記承認されたリース期間を決定するように構成された決定モジュールを備えていることを特徴とする請求項11に記載される分散ガーベッジコレクションのための装置。

【請求項14】 前記リソースアロケータは、前記リソースがそれ以前に割り当てられなかったという決定に基づいて前記リソースを割り当てるように構成された割当モジュールを備えていることを特徴とする請求項11に記載される分散ガーベッジコレクションのための装置。

【請求項15】 前記リソースアロケータは、前記リ

ソースに対応するリファレンスカウントを増加させるように構成された増加モジュールを備え、

前記リファレンスカウントは、現在、どれだけ多くのプロセスが前記リソースに対するリファレンスを持っているかを示すものであることを特徴とする請求項11に記載される分散ガーベッジコレクションのための装置。

【請求項16】 前記プロセスがある識別されたリソースに対するアクセスの終了を望んでいることを示す前記プロセスからの第2の要求を受信するように構成された第2受信モジュールと、

前記識別されたリソースに対応する前記リファレンスカウントを減少させるように構成された減少モジュールとをさらに備えたことを特徴とする請求項15に記載される分散ガーベッジコレクションのための装置。

【請求項17】 前記リソースディアルローターは、前記リソースに対する共有アクセスがもはや要求されない時に、前記リソースをリクレームするためのガーベッジコレクションプロセスを呼び出すように構成されたインボカーを備えていることを特徴とする請求項11に記載される分散ガーベッジコレクションのための装置。

【請求項18】 前記リソースディアルローターは、前記リファレンスカウントが前記識別されたリソースに、現在、いかなるプロセスもアクセスしていないことを示している時に、前記識別されたリソースの割当を解除するためのガーベッジコレクションプロセスを呼び出すように構成されたインボカーを備えていることを特徴とする請求項16に記載される分散ガーベッジコレクションのための装置。

【請求項19】 あるリース期間、あるリソースに対してアクセスすることをプロセスから要求するように構成された要求モジュールと、

前記リソースに対する共有アクセスが許可されている間、承認されたリース期間を前記プロセスから受信するように構成された受信モジュールと前記承認されたリース期間が終わりかけているが、前記リソースへのアクセスが完了していないという決定に基づいて、新たなリース期間についての他の要求を、前記プロセスに対して送信するように構成された第2送信モジュールとを備えていることを特徴とする分散ガーベッジコレクションのための装置。

【請求項20】 前記承認されたリース期間が終了する前に、前記リソースに対する共有アクセスがもはや要求されないことを示す異なる要求を送信するように構成された第2送信モジュールをさらに備えていることを特徴とする請求項19に記載される分散ガーベッジコレクションのための装置。

【請求項21】 リソースを管理するために具体化されたコンピュータで読み取り可能なコードを有するコンピュータ使用可能媒体を備えたコンピュータプログラム生産物において、

前記コンピュータ使用可能媒体は、リソースを参照し、かつ要求されたリース期間を指定するプロセスからの要求を受信するように構成された受信モジュールと、

承認されたリース期間、前記リソースへの共有アクセスを許可するように構成されたリソースアロケータと、前記承認されたリース期間を通知するように構成された通知モジュールと、

前記承認されたリース期間が終了した時に前記リソースの割当を解除するように構成されたリソースディアルローターとを備えていることを特徴とする分散ガーベッジコレクションのためのコンピュータプログラム生産物。

【請求項22】 前記リソースアロケータは、前記要求されたリース期間よりも短い期間となるように前記承認されたリース期間を決定するように構成された決定モジュールを備えていることを特徴とする請求項21に記載される分散ガーベッジコレクションのためのコンピュータプログラム生産物。

【請求項23】 前記リソースアロケータは、前記要求されたリース期間、前記リソースの大きさ、及び前記リソースのための他の承認されたリース期間の少なくとも一つに基づき前記承認されたリース期間を決定するように構成された決定モジュールを備えていることを特徴とする請求項21に記載される分散ガーベッジコレクションのためのコンピュータプログラム生産物。

【請求項24】 前記リソースアロケータは、前記リソースがそれ以前に割り当てられなかったという決定に基づいて前記リソースを割り当てるように構成された割当モジュールを備えていることを特徴とする請求項21に記載される分散ガーベッジコレクションのためのコンピュータプログラム生産物。

【請求項25】 前記リソースアロケータは、前記リソースに対するリファレンスを現在有しているプロセスの数を示すように構成されたリファレンスカウントモジュールを備えていることを特徴とする請求項21に記載される分散ガーベッジコレクションのためのコンピュータプログラム生産物。

【請求項26】 前記コンピュータ使用可能媒体は、前記プロセスがある識別されたリソースに対するアクセスの終了を望んでいることを示す前記プロセスからの第2の要求を受信するように構成された第2受信モジュールと、

前記識別されたリソースに対応する前記リファレンスカウントを減少させるように構成された減少モジュールとをさらに備えていることを特徴とする請求項25に記載される分散ガーベッジコレクションのためのコンピュータプログラム生産物。

【請求項27】 前記リソースディアルローターは、前記リソースに対する共有アクセスがもはや要求されない時に、前記リソースをリクレームするためのガーベッジ

コレクションプロセスを呼び出すように構成されたインボカーを備えていることを特徴とする請求項21に記載される分散ガーベッジコレクションのためのコンピュータプログラム生産物。

【請求項28】 前記リソースディアルケーターは、前記リファレンスカウントが前記識別されたリソースに、現在、いかなるプロセスもアクセスしていないことを示している時に、前記識別されたリソースの割当を解除するためのガーベッジコレクションプロセスを呼び出すように構成されたインボカーを備えていることを特徴とする請求項26に記載される分散ガーベッジコレクションのためのコンピュータプログラム生産物。

【請求項29】 リソースを管理するために具体化されたコンピュータで読み取り可能なコードを有するコンピュータ使用可能媒体を備えたコンピュータプログラム生産物において、

前記コンピュータ使用可能媒体は、あるリース期間、あるリソースに対してアクセスすることをプロセスから要求するように構成された要求モジュールと、

前記プロセスが前記リソースに対する共有アクセスを許可している間、承認されたリース期間を前記プロセスから受信するように構成された受信モジュールと、

前記承認されたリース期間が終わりかけているという決定に基づいて、新たなリース期間についての他の要求を、前記プロセスに対して送信するように構成された送信モジュールとを備えていることを特徴とする分散ガーベッジコレクションのためのコンピュータプログラム生産物。

【請求項30】 前記コンピュータ使用可能媒体は、前記承認されたリース期間が終了する前に、前記リソースに対する共有アクセスがもはや要求されないことを示す異なる要求を送信するように構成された第2送信モジュールをさらに備えていることを特徴とする請求項29に記載される分散ガーベッジコレクションのためのコンピュータプログラム生産物。

【請求項31】 コンピュータプロセッサと、前記コンピュータプロセッサと有効に接続されているメモリと、

前記メモリにより前記コンピュータプロセッサ中で実行するコンピュータプロセスを備え、

前記コンピュータプロセスは、リソースを参照し、かつ要求されたリース期間を指定するプロセスからの要求を受信するように構成された受信モジュールと、

承認されたリース期間、前記リソースに対する共有アクセスを許可するように構成されたリソースアロケーターと、

前記承認されたリース期間を通知するように構成された通知モジュールと、

前記承認されたリース期間が終了した時に前記リソースの割当を解除するように構成されたリソースディアルケ

ターとを備えていることを特徴とする分散ガーベッジコレクションのためのコンピュータシステム。

【請求項32】 前記リソースアロケーターは、前記要求されたリース期間よりも短い期間となるように前記承認されたリース期間を決定するように構成された決定モジュールを備えていることを特徴とする請求項31に記載される分散ガーベッジコレクションのためのコンピュータシステム。

【請求項33】 前記リソースアロケーターは、前記要求されたリース期間、前記リソースの大きさ、及び前記リソースのための他の承認されたリース期間の少なくとも一つに基づき、前記承認されたリース期間を決定するように構成された決定モジュールを備えていることを特徴とする請求項31に記載される分散ガーベッジコレクションのためのコンピュータシステム。

【請求項34】 前記リソースアロケーターは、前記リソースがそれ以前に割り当てられなかったという決定に基づいて前記リソースを割り当てるように構成された割当モジュールを備えていることを特徴とする請求項31に記載される分散ガーベッジコレクションのためのコンピュータシステム。

【請求項35】 前記リソースアロケーターは、前記リソースに対するリファレンスを現在有しているプロセスの数を示すように構成されたリファレンスカウントモジュールを備えていることを特徴とする請求項31に記載される分散ガーベッジコレクションのためのコンピュータシステム。

【請求項36】 前記コンピュータプロセスは、前記プロセスがある識別されたリソースに対するアクセスの終了を望んでいることを示す前記プロセスからの第2の要求を受信するように構成された第2受信モジュールと、前記識別されたリソースに対応する前記リファレンスカウントを減少させるように構成された減少モジュールとをさらに備えていることを特徴とする請求項35に記載される分散ガーベッジコレクションのためのコンピュータシステム。

【請求項37】 前記リソースディアルケーターは、前記リソースに対する共有アクセスがもはや要求されない時に、前記リソースをリクレームするためのガーベッジコレクションプロセスを呼び出すように構成されたインボカーを備えていることを特徴とする請求項31に記載される分散ガーベッジコレクションのためのコンピュータシステム。

【請求項38】 前記リソースディアルケーターは、前記リファレンスカウントが前記識別されたリソースに、現在、いかなるプロセスもアクセスしていないことを示している時に、前記識別されたリソースの割当を解除するためのガーベッジコレクションプロセスを呼び出すように構成されたインボカーを備えていることを特徴とする請求項36に記載される分散ガーベッジコレクシ

ンのためのコンピュータシステム。

【請求項39】 コンピュータプロセッサと、  
前記コンピュータプロセッサと有効に接続されている  
メモリーと、

前記メモリーにより前記コンピュータプロセッサ中で  
実行するコンピュータプロセスを備え、

前記コンピュータプロセスは、あるリース期間、あるリ  
ソースに対してアクセスすることをプロセスから要求す  
るよう構成された要求モジュールと、

前記プロセスが前記リソースに対する共有アクセスを許  
可している間、承認されたリース期間を前記プロセスか  
ら受信するよう構成された受信モジュールと、

前記承認されたリース期間が終わりかけているという決  
定に基づいて、新たなリース期間についての他の要求  
を、前記プロセスに対して送信するよう構成された送  
信モジュールとを備えていることを特徴とする分散ガー  
ベッジコレクションのためのコンピュータシステム。

【請求項40】 前記コンピュータプロセスは、前記承認  
されたリース期間が終了する前に、前記リソースに対  
する共有アクセスがもはや要求されないことを示す異な  
る要求を送信するよう構成された第2送信モジュール  
をさらに備えていることを特徴とする請求項39に記載  
される分散ガーベッジコレクションのためのコンピュ  
ータシステム。

【請求項41】 複数のプラットフォームを有する分散  
処理システムにおいて、サーバープラットフォームで実  
行される以下の工程、すなわち、(a) システムリソ  
ースに対するリファレンス及び要求されたリース期間を  
含むリクエストプラットフォームからの要求であつて、

前記対応するリクエストプラットフォームが前記要求  
中において前記要求された期間、前記リファレンスされ  
たシステムリソースへのアクセスをシークすることを示  
している要求を受信する工程と、(b) (i) 前記  
要求された各リース期間に回答して設定される、対応す  
る承認期間であって、要求に関連する前記リファレンス  
された各システムリソースが前記プラットフォームによ  
ってアクセスされることが可能となる期間を示している  
承認期間を設定すること、及び(ii) 対応するリフ  
ァレンスカウンターであって、要求の受信によって決定  
される前記対応するシステムリソースがアクセスしてい  
る前記プラットフォームの数を示しているリファレンス  
カウンターを増加させること、により前記リファレンス  
された各システムリソースを割り当てる工程と、(c)

関連する要求に回答して、関連する要求中でリファ  
レンスされた前記リソースのための前記承認期間を指定す  
る前記リクエストプラットフォームに対する応答を送  
信する工程と、(d) (i) リファレンスされた各  
システムリソースのための前記承認期間が終了した時、  
又は(ii) 前記対応するリファレンスカウンター

が、いかなるリクエストプラットフォームも、現在、  
前記リファレンスされた各システムリソースに対するア  
クセスをシークしていないことを示す状態である  
時に、前記リファレンスされた各システムリソースの割  
当を解除するためのガーベッジコレクションプロセスを  
起動する工程とを備えたことを特徴とする分散ガーベ  
ッジコレクションのためのプラットフォーム間に分散した  
リソースの割り及び割当解除方法。

【請求項42】 (e) 前記システムリソースに対す  
るリファレンスを含み、かつ前記リクエストプラット  
フォームが前記システムリソースに対するアクセスの終  
了を要求することを示している前記リクエストプラッ  
トフォームからの新しい要求を受信する工程と、(f)

前記新しい要求において指定された前記リファレンス  
された各システムリソースに対応する前記リファレンス  
カウンターを減少させる工程とをさらに備えていること  
を特徴とする請求項41に記載される分散ガーベッジコ  
レクションのためのプラットフォーム間に分散したリソ  
ースの割り及び割当解除方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、一般に、コンピ  
ュータシステムのためのガーベッジコレクションに関  
し、更に詳しくは、リファレンスと結合し、あるいは関  
連しているリソースを収集するための耐故障性の分散ガ  
ーベッジコレクションの方法に関する。

【0002】

【従来の技術】適切なリソース管理は、コンピュータの  
効率性及び効果的使用のための重要な観点である。一般  
に、リソース管理には、適切な時、例えば、要求者がも  
はやリソースを要求しない時にリソースの割り当てを解  
除することのみならず、要求に応じてリソース（例え  
ば、メモリー）を割り当てることも含まれる。一般に、  
リソースには、コンピュータ内で実行する演算の構成要  
素（例えば、アプリケーション、プログラム、アプレッ  
トなど）によって引用されるデータが含まれる。

【0003】実際には、コンピュータ上で実行している  
アプリケーションがリソースを参照するためにシークす  
る時には、コンピュータは、アプリケーションが適切に  
それらを参照することができるように、最初にリソース  
を割り当てるか、あるいは指定しなければならない。ア  
プリケーションが、もはやリソースを参照しないとき  
は、コンピュータは、再使用のためにそのリソースの割  
り当てを解除するか、あるいはリクレームすることがで  
きる。コンピュータ内では、各リソースは、それによっ  
てそのリソースが引用可能な唯一の「ハンドル」を持っ  
ている。そのハンドルは、アドレス、配列インデック  
ス、特有の値、ポインター等のような、様々な方法によ  
って与えられる。

【0004】リソース管理は、1台のコンピュータにと

っては比較的単純である。なぜならば、アプリケーションがもはやそれらを参照しない時、あるいは停電後のように、リソースがリクレーンされることのできる時を示す事象は、決定することが容易だからである。多数のコンピュータを接続した分散システムのためのリソース管理は、数台の異なるコンピュータ内のアプリケーションが同一のリソースを使っていることもあるので、より困難である。

【0005】分散システムの分断は、不適切かつ先走ったリソースのリクレーン、あるいはリソースのリクレーンの失敗の原因となる。例えば、分散システムにおいて異なるコンピュータ上で実行されている多数のアプリケーションは、他の機械上に配置されているリソースを参照するかもしれない。もし、リソースが配置されているコンピュータと、それらのリソースを参照しているアプリケーションとの間の接続が途切れるならば、コンピュータは、早まってそのリソースをリクレーンすることもあり得る。代わりに、コンピュータは、アプリケーションがそのリソースにアクセスすることに失敗してから長期間経過しているにもかかわらず、永続的にそのリソースを保持することもありうる。

【0006】これらの問題は、ネットワークリソースを管理するためのシステムの発展をもたらし、その一つは、「分散ガーベッジコレクション」として知られている。この用語は、ネットワーク中の異なるコンピュータ上で実行されている一つのアプリケーションあるいはアプリケーション群によって用いられているリソースを自動的に管理する、分散システムのための言語又は実行時間システムによって与えられる機能を表している。

【0007】一般に、ガーベッジコレクションは、リソースがもはやアプリケーションのいかなる部分によっても引用されない時に、リソースは、将来の使用のために開放されるという概念を用いている。分散ガーベッジコレクションは、いかなるコンピュータ上のアプリケーションもリソースを参照しない時にリソースをリクレーンすることにより、この概念を、分散コンピューティングの領域に拡張する。

【0008】分散ガーベッジコレクションは、割り当てられたリソースとそれらのリソースのリファレンスとの間の健全性を保持しなければならない。換言すれば、そのシステムは、ネットワーク中のいずれかのコンピュータ上で実行されているアプリケーションがそのリソースの参照を続けている時には、リソースの割り当てを解除したり、あるいはリソースを開放することが許されない。「リファレンスの健全性」と称するこのリファレンスとリソースの結合は、リファレンスが参照するリソースへのアクセスを常に承認することことを保証するものではない。例えば、ネットワークの故障は、そのようなアクセスを不可能にする。しかしながら、健全性は、もし、そのリファレンスがいずれかのリソースへの

アクセスを得るために用いられたならば、そのリソースは、そのリファレンスが最初に与えられたリソースと同一のリソースであるということを保証するものである。

【0009】ガーベッジコレクションが用いられる分散システムは、また、有限の未来におけるある時点においてもはや引用されなくなったリソースをリクレーンしなければならない。換言すれば、そのシステムは、「記憶漏れ」に対する保証を与えなければならない。記憶漏れは、全てのアプリケーションがリソースに対するリファレンスを落としたが、例えば、いくつかのアプリケーションが未だそのリソースを参照しているという誤った決定をしたために、システムが、再使用のためにリソースをリクレーンすることに失敗した時に生ずる。

【0010】リファレンスの健全性の故障と記憶漏れは、しばしば、リソースを引用しているアプリケーションとそれらのリソースの割当及び割当解除を管理しているガーベッジコレクションシステムとの間の分断の原因となる。例えば、リソースを参照しているアプリケーションとそのリソースを管理しているガーベッジコレクションシステムとの間のネットワーク結合の分断は、ガーベッジコレクションシステムが、リソースのリクレーンをすべきか否か及びリソースをいつリクレーンすべきかを決定することを妨げるかもしれない。代わりにして、ガーベッジコレクションシステムは、アプリケーションが予め定められた時間内にリソースにアクセスしなかったので、そのリソースを収集してもよいということを誤って決定するかもしれない。

【0011】分散ガーベッジコレクションのメカニズムが記憶漏れを起こすことなくリファレンスの健全性を保持するということを保証することを試みることにより、その分散ガーベッジコレクションのメカニズムを改善するために、多くのテクニックが用いられている。一つのありふれたアプローチは、リファレンスカウントの形式を用いている。そこでは、カウントは、各リソースを参照しているアプリケーションの数を保持している。リソースのカウントがゼロになるとき、ガーベッジコレクションシステムは、そのリソースをリクレーンすることができる。

【0012】しかしながら、そのようなリファレンスカウントの機構は、リソースが対応するリファレンスカウンターと共に作られる時にのみ動作する。この場合、ガーベッジコレクションシステムは、追加のアプリケーションがそのリソースを参照するときには、そのリソースのリファレンスカウントを増加させ、あるアプリケーションがもはやそのリソースを参照しない時には、カウントを減少させる。

【0013】しかしながら、リファレンスカウントの機構は、特に、分散システムにおいて故障を生ずるという問題に直面している。そのような故障は、リソースはもはや引用されないということをガーベッジコレクション



システムに知らせるメッセージの伝達を妨げる、コンピュータもしくはアプリケーションの故障、又はネットワークの故障という形をとる。もし、メッセージがネットワークが分断しているために伝達されないならば、不要情報収集システムは、いつリソースをリクレーンすべきかわからない。

【0014】そのような故障を避けるために、いくつかのありふれたリファレンスカウントの機構には、「キープアライブ」メッセージが含まれている。これは、また、「ピンバック」とも称されている。この機構によれば、ネットワーク中のアプリケーションは、リソースを監視しているガーベッジコレクションシステムに対してメッセージを送り、そのアプリケーションがなお通信可能であることを示す。これらのメッセージは、ガーベッジコレクションシステムがリソースに対するリファレンスを落とすのを妨げる。

【0015】そのような「キープアライブ」メッセージの受信の失敗は、ガーベッジコレクションシステムが、あるリソースのためのリファレンスカウントを減少させることができること、そしてすなわち、そのカウントがゼロになるときは、ガーベッジコレクションシステムは、そのリソースをリクレーンするかもしれないということの意味している。しかしながら、これは、ネットワークが故障しているために、「キープアライブ」メッセージを受け取ることに失敗したことから、リファレンスカウントがゼロへ到達し、それに続いて、リソースの早まったリクレーンが生ずる。これは、リファレンスの保全性の必要条件を乱すものである。

【0016】ガーベッジコレクションシステムにおけるリファレンスの保全性の問題を解決するために提案された他の方法は、リファレンスカウントのみならず、リソースを参照するコンピュータの各構成要素に対応する識別子もまた保持することである。A. Birrell 他、「ネットワークの目的物のための分散ガーベッジコレクション」、No. 116、デジタル システムズ リサーチ センター、1993年12月15日参照。

【0017】この方法は、リファレンスカウントの機構と同一の問題に悩まされている。さらに、この方法は、分散形システム内で不必要な通信を増加させ、かつ記憶のための必要条件（すなわち、各リソースを参照するアプリケーションに対応する識別子のリスト）を追加するオーバーヘッドを付加した唯一の識別子を、各リソースを参照するコンピュータの各構成要素に与えることを必要とする。

【0018】

【発明が解決しようとする課題】本発明が解決しようとする課題は、記憶条件の追加や不必要な通信の増大を招くことなく、リファレンスの保全性が維持され、かつ記憶漏れを起こすことのない分散ガーベッジコレクションのためのリソースの管理方法、装置、コンピュータプロ

グラム生産物、コンピュータシステム並びにプラットフォーム間に分散したリソースの割当及び割当解除方法を提供することにある。

【0019】

【課題を解決するための手段】上記課題を解決するために、本発明に係る分散ガーベッジコレクションのためのリソースの管理方法は、リソースを参照し、かつ要求されたリース期間を指定するプロセスからの要求を受信する受信工程と、承認されたリース期間、前記リソースに対する共有アクセスを許可する許可工程と、前記プロセスに前記承認されたリース期間を通知する通知工程と、前記承認されたリース期間が終了した時に前記リソースの割当を解除する割当解除工程とを備えていることを要旨とするものである。

【0020】また、本発明に係る他の分散ガーベッジコレクションのためのリソースの管理方法は、あるリース期間、あるリソースに対してアクセスすることをプロセスから要求する工程と、前記リソースに対する共有アクセスが許可されている間、承認されたリース期間を前記プロセスから受信する工程と、前記承認されたリース期間が終わりかけているが、前記リソースへのアクセスが完了していないという決定に基づいて、新たなリース期間の要求を前記プロセスに対して送信する工程とを備えていることを要旨とするものである。

【0021】本発明に係る分散ガーベッジコレクションのための装置は、リソースを参照し、かつ要求されたリース期間を指定するプロセスからの要求を受信するように構成された受信モジュールと、承認されたリース期間、前記リソースへの共有アクセスを許可するように構成されたリソースアロケータと、前記プロセスに前記承認されたリース期間を通知するように構成された通知モジュールと、前記承認されたリース期間が終了した時に前記リソースの割当を解除するように構成されたリソースデアロケータとを備えていることを要旨とするものである。

【0022】また、本発明に係る他の分散ガーベッジコレクションのための装置は、あるリース期間、あるリソースに対してアクセスすることをプロセスから要求するように構成された要求モジュールと、前記リソースに対する共有アクセスが許可されている間、承認されたリース期間を前記プロセスから受信するように構成された受信モジュールと前記承認されたリース期間が終わりかけているが、前記リソースへのアクセスが完了していないという決定に基づいて、新たなリース期間についての他の要求を、前記プロセスに対して送信するように構成された第2送信モジュールとを備えていることを要旨とするものである。

【0023】本発明に係る分散ガーベッジコレクションのためのコンピュータプログラム生産物は、リソースを管理するために具体化されたコンピュータで読み取り可

13

能なコードを有するコンピュータ使用可能媒体を備えたコンピュータプログラム生産物において、前記コンピュータ使用可能媒体は、リソースを参照し、かつ要求されたリース期間を指定するプロセスからの要求を受信するように構成された受信モジュールと、承認されたリース期間、前記リソースへの共有アクセスを許可するように構成されたリソースアロケータと、前記承認されたリース期間を通知するように構成された通知モジュールと、前記承認されたリース期間が終了した時に前記リソースの割当を解除するように構成されたリソースディアルケータとを備えていることを要旨とするものである。

【0024】また、本発明に係る他の分散ガーベッジコレクションのためのコンピュータプログラム生産物は、リソースを管理するために具体化されたコンピュータで読み取り可能なコードを有するコンピュータ使用可能媒体を備えたコンピュータプログラム生産物において、前記コンピュータ使用可能媒体は、あるリース期間、あるリソースに対してアクセスすることをプロセスから要求するように構成された要求モジュールと、前記プロセスが前記リソースに対する共有アクセスを許可している間、承認されたリース期間を前記プロセスから受信するように構成された受信モジュールと、前記承認されたリース期間が終わりかけているという決定に基づいて、新たなリース期間についての他の要求を、前記プロセスに対して送信するように構成された送信モジュールとを備えていることを要旨とするものである。

【0025】本発明に係る分散ガーベッジコレクションのためのコンピュータシステムは、コンピュータプロセッサと、前記コンピュータプロセッサと有効に接続されているメモリと、前記メモリにより前記コンピュータプロセッサ中で実行するコンピュータプロセスを備え、前記コンピュータプロセスは、リソースを参照し、かつ要求されたリース期間を指定するプロセスからの要求を受信するように構成された受信モジュールと、承認されたリース期間、前記リソースに対する共有アクセスを許可するように構成されたリソースアロケータと、前記承認されたリース期間を通知するように構成された通知モジュールと、前記承認されたリース期間が終了した時に前記リソースの割当を解除するように構成されたリソースディアルケータとを備えていることを要旨とするものである。

【0026】また、本発明に係る他の分散ガーベッジコレクションのためのコンピュータシステムは、コンピュータプロセッサと、前記コンピュータプロセッサと有効に接続されているメモリと、前記メモリにより前記コンピュータプロセッサ中で実行するコンピュータプロセスを備え、前記コンピュータプロセスは、あるリース期間、あるリソースに対してアクセスすることをプロセスから要求するように構成された要求モ

14

ジュールと、前記プロセスが前記リソースに対する共有アクセスを許可している間、承認されたリース期間を前記プロセスから受信するように構成された受信モジュールと、前記承認されたリース期間が終わりかけているという決定に基づいて、新たなリース期間についての他の要求を、前記プロセスに対して送信するように構成された送信モジュールとを備えていることを要旨とするものである。

【0027】さらに、本発明に係る分散ガーベッジコレクションのためのプラットフォーム間に分散したリソースの割当及び割当解除方法は、複数のプラットフォームを有する分散処理システムにおいて、サーバプラットフォームで実行される以下の工程、すなわち、(a) システムリソースに対するリファレンス及び要求されたリース期間を含むリクエストプラットフォームからの要求であって、前記対応するリクエストプラットフォームが前記要求中において前記要求された期間、前記リファレンスされたシステムリソースへのアクセスをシークすることを示している要求を受信する工程と、(b)

(i) 前記要求された各リース期間に回答して設定される、対応する承認期間であって、要求に関連する前記リファレンスされた各システムリソースが前記プラットフォームによってアクセスされることが可能となる期間を示している承認期間を設定すること、及び(ii)

対応するリファレンスカウンターであって、要求の受信によって決定される前記対応するシステムリソースがアクセスしている前記プラットフォームの数を示しているリファレンスカウンターを増加させること、により前記リファレンスされた各システムリソースを割り当てる工程と、(c) 関連する要求に回答して、関連する要求中でリファレンスされた前記リソースのための前記承認期間を指定する前記リクエストプラットフォームに対する応答を送信する工程と、(d) (i) リファレンスされた各システムリソースのための前記承認期間が終了した時、又は(ii) 前記対応するリファレンスカウンターが、いかなるリクエストプラットフォームも、現在、前記リファレンスされた各システムリソースに対するアクセスをシークしていないということを示す状態である時に、前記リファレンスされた各システムリソースの割当を解除するためのガーベッジコレクションプロセスを起動する工程とを備えたことを要旨とするものである。

【0028】上記構成を有する本発明に係るリソースの管理方法によれば、リファレンスの保安全性は、ある一定の期間、すなわち、分散システム中のパーティ、例えば、リソースに対するリファレンスを保持しているアプリケーション及びそのリソースを管理しているガーベッジコレクションシステムが、そのリソースとそのリソースに対するリファレンスとが保証されることについて合意している期間、リソースをリースすることによって、

大きな記憶漏れを伴うことなく保証される。

【0029】そのリース期間が終了したときには、そのリソースに対するリファレンスが継続するという保証は、消滅し、ガーベッジコレクションシステムにそのリソースをリクレーンすることを許可する。なぜならば、そのリソースに対するリファレンスを保持しているアプリケーションと、そのリソースを管理しているガーベッジコレクションシステムは、有限の保証されたリース期間について合意しており、双方とも、いつそのリースが終了するか、すなわちいつその保証が終了するかを知ることができるからである。これにより、リファレンスのリース期間中はリファレンスの健全性が保証され、ネットワークエラーに起因して発生するリソースの開放失敗を回避することが可能となる。

【0030】

【発明の実施の形態】以下に、添付した図面の記載に基づいて、本発明の実施の形態を詳細に説明する。可能な限り、同一又は類似の部分について言及する図面及び以下の記述全体を通して、同一の参照番号を使用する。

【0031】本発明は、ありふれた分散処理システムの構成によって組織されたコンピュータを用いて実行することができる。しかしながら、本発明を実行するための構成と手順は、ありふれたものではない。なぜならば、それらは、リファレンスの健全性を保証し、記憶漏れを排除する分散ガーベッジコレクション機構を与えるものだからである。

【0032】(A. 総括) 分散処理システム中にある各コンピュータに配置されている方法呼出要素(以下、「M I 要素」という)が、本発明に係る分散ガーベッジコレクション機構を実行する。M I 要素は、好ましくは、プログラム言語 J A V A (登録商標)によって書かれた多くのソフトウェアモジュールからなっている。

【0033】一般に、分散処理システム中のアプリケーションが、分散リソースに対するリファレンスを、ネームの索引あるいは他の方法によって、ある他のコールに対するリターン値として取得し、そのリソースへのアクセスをシークする時はいつでも、そのアプリケーションは、そのリソースに対し、あるいはそのリソースを管理するM I 要素に対してコールを行う。

【0034】マネージングM I 要素と呼ばれるこのM I 要素は、そのリソースに対して突出しているリファレンス数のトラックをキープする。あるリソースに対するリファレンスの数がゼロの時は、マネージングM I 要素は、そのリソースをリクレーンすることができる。あるリソースに対するリファレンス数のカウントは、一般に、「リファレンスカウント」と呼ばれており、リファレンスカウントを増加させるコールは、「ダーティーコール」と称される。

【0035】アプリケーションが、もはや分散リソースを要求しない時は、アプリケーションは、そのリソース

に対しあるいはマネージングM I 要素に対して、異なるコールを送信する。このコールの受信に基づき、マネージングM I 要素は、そのリソースのリファレンスカウントを減少させる。このリファレンスを落とすためのコールは、「クリーンコール」と称される。

【0036】本発明に係る実施の形態によれば、ダーティーコールには、リース期間と呼ばれる、そのリソースに対するリファレンスのための要求された時間間隔を含めることができる。ダーティーコールの受信に基づき、マネージングM I 要素は、リースが承認された期間を含んでいるリターンコールを送信する。すなわち、マネージングM I 要素は、突出したリファレンスの数と同様に、それらのリファレンスのためのリース期間もトラックする。その結果、あるリソースに対するリファレンスカウントがゼロになった時、あるいはそのリソースのためのリース期間が終了した時には、マネージングM I 要素は、そのリソースをリクレーンすることができる。

【0037】(B. 手順) M I 要素中のアプリケーションコールプロセッサは、図1に図示されたアプリケーションコール手順100のステップを実行する。マネージングM I 要素中のサーバーコールプロセッサは、それぞれ図2～4に図示される、手順200、300及び400のステップを実行する。マネージングM I 要素のガーベッジコレクターは、サーバーコールプロセッサからの命令に従って、それ以前にリファレンスと結合していたリソースをリクレーンするために通常の手順を実行する。従って、ガーベッジコレクターの通常の手順については、説明を省略する。

【0038】(1. アプリケーションコールプロセッサ) 図1は、分散処理システム中に配置されている同一又は他のM I 要素によって管理されるリソースに対するリファレンスのためのアプリケーションの要求を取り扱うために、M I 要素のアプリケーションプロセッサが使用する手順100のフロー図を示したものである。

【0039】アプリケーションがあるリソースに対するリファレンスを取得した後、アプリケーションコールプロセッサは、リソースのリファレンス及び要求されたリース期間を含むダーティーコールを、そのリソースのためのマネージングM I 要素に対して送信する(ステップ110)。ダーティーコールは、リソースそれ自体に対してあるいはマネージングM I 要素に対して向けられる。

【0040】次いで、アプリケーションコールプロセッサは、マネージングM I 要素からのリターンコールを待ち、そして受信する(ステップ120)。そのリターンコールには、ダーティーコールのリファレンスとそのリソースと結合するであろうということをマネージメントM I 要素が保証する期間、すなわち、承認されたリース期間が含まれている。換言すれば、マネージングM I 要素は、その承認期間、ダーティーコールのリファレン

スに対応するリソースを収集しないことに同意する。もし、マネージングMI要素が承認期間を与えないか、あるいはリースの要求を拒絶するならば、アプリケーションコールプロセッサは、承認期間を受信するまで、別のダーティーコールを送信しなければならない。

【0041】アプリケーションコールプロセッサは、アプリケーションによるそのリファレンスの使用を監視し、そのアプリケーションが、アプリケーションコールプロセッサに対し、そのリファレンスがもはや要求されないということを明確に知らせた時、又は、そのアプリケーションプロセッサが、自分自身でその決定を下したの時のいずれかの時（ステップ130）には、アプリケーションコールプロセッサは、マネージングMI要素に対してクリーンコールを送信する（ステップ140）。ダーティーコールに用いられた方法と同様の方法により、クリーンコールは、リファレンスされたリソースに向けられ、マネージングMI要素は、クリーンコールを処理する。続いて、アプリケーションコールプロセッサは、アプリケーションによって使用されているリファレンスのリストからそのリファレンスを削除する（ステップ150）。

【0042】もし、アプリケーションが、いまだリファレンスをやめていない（ステップ130）が、アプリケーションコールプロセッサが、そのリファレンスのための承認期間が終了しそうであると判断する場合（ステップ160）には、次に、アプリケーションコールプロセッサは、そのリソースに対するリファレンスが、アプリケーションに代わってマネージングMI要素によって保持されているということを保証するために、ステップ110とステップ120を繰り返す。

【0043】（2. サーバーコールプロセッサ）MI要素のサーバーコールプロセッサは、3つの主要な手順を実行する。すなわち、（1）ダーティーコールを処理すること、（2）入ってくるクリーンコールを処理すること、及び（3）適切なときに、リソースをリクレームするためのガーベッジコレクションサイクルを起動させること、である。

【0044】（i. ダーティーコール）図2は、MIソフトウェア要素が管理するリファレンスされたリソースに対する要求、すなわちダーティーコールを処理するために、MI要素のサーバーコールプロセッサが使用する手順200のフロー図である。これらの要求は、分散処理システム中にあるMI要素のアプリケーションコールプロセッサからくるものであり、そのアプリケーションコールプロセッサには、要求を処理するサーバーコールプロセッサと同じMI要素中のアプリケーションコールプロセッサも含まれている。

【0045】最初に、サーバーコールプロセッサは、ダーティーコールを受信する（ステップ210）。次いで、サーバーコールプロセッサは、受入可能な承認期

間を決定する（ステップ220）。承認期間は、要求されたリース期間、あるいは他の期間と同一でもよい。サーバーコールプロセッサは、要求されたリソースの量及び同一のリソースに対してそれ以前に承認された他の多くの承認期間を含む多数の条件に基づいて適切な承認期間を決定する。

【0046】リソースがダーティーコールのリファレンスにより未だ割り当てられていないとサーバーコールプロセッサが判断した時（ステップ230）には、サーバーコールプロセッサは、要求されたリソースを割り当てる（ステップ240）。

【0047】次いで、サーバーコールプロセッサは、ダーティーコールのリファレンスに対応するリファレンスカウントを増加させ（ステップ250）、リファレンス・リソース結合のための受入可能な承認期間を設定し（ステップ260）、アプリケーションプロセッサに対し承認期間と共にリターンコールを送信する（ステップ270）。このようにして、サーバーコールプロセッサは、その制御下にあるリソースに対するリファレンスに注意しながら、入ってくるダーティーコールを制御する。

【0048】アプリケーションは、現在のリースが終了する前に、拡張要求を伴うダーティーコールを送信することによって、リースを延長することができる。手順200において示したように、リースを拡張するための要求は、リースのための最初の要求と全く同様に扱われる。拡張は、単に、リファレンスカウントがゼロにならない限り、ある付加的な期間、リソースはリクレームされないということを意味するにすぎない。

【0049】（ii. クリーンコール）MI要素のサーバーコールプロセッサは、また、アプリケーションコールプロセッサから入ってくるクリーンコールも処理する。分散処理システム中のアプリケーションがもはやリソースに対するリファレンスを要求しない時は、アプリケーションは、そのリソースが再使用のためにリクレームされるように、そのリファレンスのためのリソースを管理するMI要素に知らせる。図3は、MI要素のサーバーコールプロセッサがクリーンコールを処理するために使用するステップを有する手順300のフロー図を示している。

【0050】サーバーコールプロセッサが、MI要素が管理するリソースに対するリファレンスと共にクリーンコールを受信した時（ステップ310）は、サーバーコールプロセッサは、対応するリファレンスカウントを減少させる（ステップ320）。クリーンコールは、そのリソースを監視し、かつそのコールを処理するための手順300を実行するとサーバーコールプロセッサと共に、そのリソースに送信される。続いて、サーバーコールプロセッサは、受信を承認するために、クリーンコールを送信しているMI要素に対してリターンコー

ルを送信する(ステップ330)。本発明に係る実施の形態によれば、リファレンスを落とすためのクリーンコールは拒絶されないが、承認されなければならない。

【0051】(iii. ガーベッジコレクション) サーバーコールプロセッサは、また、もはやリソースに対してリファレンスがなされないか、あるいはそのリソースのための合意されたリース期間が終了したかのいずれかを決定したそのリソースをリクレームするために、ガーベッジコレクションサイクルを起動させる。図4に示した手順400は、サーバーコールプロセッサがガーベッジコレクションサイクルを起動するために使用するステップのフロー図を示している。

【0052】サーバーコールプロセッサは、リファレンスカウント及び承認されたリース期間を監視し、MI要素によって管理されているあるリソースについてリファレンスカウントがゼロであるか否か、あるいはあるリファレンスのための承認期間が終了したか否かを決定する(ステップ410)。いずれかの条件が存在するときは、サーバーコールプロセッサは、そのリソースのガーベッジコレクションを起動する(ステップ420)。そうでない場合は、サーバーコールプロセッサは、リファレンスカウントと承認されたリース期間の監視を続行する。

【0053】(C. コールフロー) 図5は、分散処理システム内にあるMI要素間のコールの流れを模式的に表した図である。マネージングMI要素525は、リソース530に対するリファレンスを監視することによって、リソース530を管理する(ガーベッジコレクト505参照)。マネージングMI要素525は、リソースを管理するので、マネージングMI要素525のサーバーコールプロセッサは、ここに記載されたコールフローの処理を実行する。

【0054】図5は、また、アプリケーション510及び540が、それぞれ、対応するMI要素515及び545を有していることを示している。各アプリケーション510及び540は、それぞれ、リソース530の内の一つに対するリファレンスを取得し、リファレンスが対応するリソースと結合するように、リソース530の内の一つに対するアクセスを取得するためにシークする。

【0055】アクセスを取得するために、アプリケーション510及び540は、MI要素525に対して、それぞれ、ダーティーコール551及び571を送信するために、それぞれ、対応するMI要素515及び545を呼び出す。MI要素515及び545は、マネージングMI要素525のような他のMI要素によって管理されるリソース530に対してアクセスするためのアプリケーションの要求を処理するので、MI要素515及び545のアプリケーションコールプロセッサは、ここに記載されたコールフローの処理を実行する。

【0056】ダーティーコール551及び571に回答して、マネージングMI要素525は、各MI要素515及び545に対して、それぞれ、リターンコール552及び572を送信する。ダーティーコールには、ダーティーコール551及び571のリファレンスのための承認されたリース期間を含んでいる。

【0057】同様に、図5は、また、マネージングMI要素525に対して、それぞれ、クリーンコール561及び581を送信しているMI要素515及び545を示している。クリーンコール561及び581は、アプリケーション510及び540が、それぞれ、クリーンコール561及び581で指定されるリソースに対するアクセスをもちや要求しないということをマネージングMI要素525に知らせる。マネージングMI要素525は、クリーンコール561及び581に対して、それぞれ、リターンコール562及び582で応答する。リターンコール562及び582は、それが、受信したクリーンコール561及び581に対するMI要素525からの単なる承認であるという点において、リターンコール552及び572とは異なる。

【0058】アプリケーション510及び540の双方とも、同一リソースに対するアクセスを要求するかもしれない。例えば、アプリケーション540がそれ以前に「リソース(1)」へのアクセスが承認されていた一方で、アプリケーション510が、「リソース(1)」に対するアクセスを要求することもある。MI要素525は、合意されたリース期間、アプリケーション510及び540の双方がそのリソースを使用できるようにすることによってこのような状況を処理する。すなわち、MI要素525は、アプリケーション510及び540の双方とも、そのリソースに対するリファレンスを落とすか、あるいは最新の合意された期間が終了したかのいずれかの事象が最初に生ずるまで、「リソース(1)」をリクレームするためにガーベッジコレクションサイクルを起動させない。

【0059】1より多いアプリケーションに対し、同一のリソースに対して同時にアクセスすることを許可することによって、本発明は、また、アプリケーションが、リソースに対するリファレンスを落とすように、マネージングMI要素に対してクリーンコールを送信した後、そのアプリケーションがそのリソースにアクセスすることを許可する。これは、そのリソースが他のアプリケーションによっていまなおリファレンスされているか、あるいはそのリファレンスのリースが未だに終了しておらず、それによりマネージングMI要素525が未だにそのリソースをリクレームしていないために生ずる。しかしながら、そのリソースは、有限の期間が経過した後、すなわち、いかなるアプリケーションもリースしていない時、あるいは最新のリースが終了した時に、リクレームされることになる。

【0060】(D. M1要素)図6は、本発明の実施の形態に係るM1要素600のモジュールのブロック図を示している。M1要素600には、監視されている各リファレンスのためのリファレンス要素605、アプリケーションコールプロセッサ640、サーバーコールプロセッサ650及びガーベッジコレクター660を含めることができる。

【0061】リファレンス要素605は、好ましくは、リファレンスデータポーション610に備えられた表又はこれに相当する構造、リファレンスカウント620及び承認期間レジスター630で構成する。M1要素600は、対応するリソースをリクレーンするためにガーベッジコレクター660をいつ起動させるかを決定するために、対応するリファレンスデータポーション610で指定される各リファレンスのための、リファレンスカウント620と承認期間630を使用する。

【0062】アプリケーションコールプロセッサ640は、図1に示す手順100のステップを実行するソフトウェアモジュールである。サーバーコールプロセッサ650は、図2～4に示す手順200、300及び400のステップを実行するソフトウェアモジュールである。ガーベッジコレクター660は、上述したように、サーバーコールプロセッサ650からの指令に回答してリソースをリクレーンするソフトウェアモジュールである。

【0063】(E. 分散処理システム)図7は、本発明を実施するために用いられる分散処理システム50を図示したものである。図7において、分散処理システム50は、ネットワーク雲55によって表現されるネットワーク配列中に接続された3つの独立で、かつ異質なプラットフォーム700、800及び900を含んでいる。図7中で、雲55によって表現されるネットワーク配列の構成及びプロトコルは、プラットフォーム700、800及び900間の情報伝達を許すものである限り、重要なことではない。

【0064】加えて、ちょうど3つのプラットフォームを使用していることは、単なる例示にすぎないものであり、本発明が、非常に多くのプラットフォームを使用することを制限するものではない。さらに、特定のネットワーク構造は、本発明にとって重要なことではない。例えば、本発明に従って使用することができる他のネットワーク構造では、他の全てのプラットフォームが接続されるネットワークコントローラーとして、1つのプラットフォームを使用する。

【0065】分散処理システム50の実施の形態において、プラットフォーム700、800及び900は、それぞれ、プロセッサ710、810及び910、並びにメモリー750、850及び950を含んでいる。各プロセッサ710、810及び910には、それぞれ、アプリケーション720、820及び920、オペ

レーティングシステム740、840及び940、並びにM1要素730、830及び930が含まれている。

【0066】アプリケーション720、820及び920は、本発明を実行するために以前に書かれかつ修正されたプログラム、あるいは本発明によって提供されるサービスを利用するために特別に書かれたプログラムのいずれでもよい。アプリケーション720、820及び920は、本発明に従って実行されるオペレーションを呼び出す。

【0067】M1要素730、830及び930は、図6を参照しながら前述したM1要素600に対応している。

【0068】オペレーティングシステム740、840及び940は、それぞれ、対応するプロセッサ710、810及び910と連結された標準的なオペレーティングシステムである。プラットフォーム700、800及び900は、異質なものとすることができる。例えば、プラットフォーム700は、プロセッサ710として、サン・マイクロシステムズ社製ウルトラスパークマイクロプロセッサを有し、オペレーティングシステム740としてソラリスを使用する。

【0069】プラットフォーム800は、プロセッサ810として、シリコングラフィックス社製MIPSマイクロプロセッサを有し、オペレーティングシステム840としてユニックスを使用する。最後に、プラットフォーム900は、プロセッサ910としてインテル社製ペンティアムマイクロプロセッサを有し、オペレーティングシステム940としてマイクロソフトウィンドウズ95を使用する。本発明は、これに限定されるものではなく、さらに同質のプラットフォームを収容することもできる。

【0070】サン、サン・マイクロシステムズ、ソラリス、ジャバ及びサンロゴは、米国及び他の国におけるサン・マイクロシステムズ社の商標又は登録商標である。ウルトラスパーク及び他の全てのスパーク商標は、許諾に基づいて使用されており、米国及び他の国におけるスパークインターナショナル社の商標である。スパーク商標を付した製品は、サン・マイクロシステムズ社によって開発された構造を基礎に置いている。

【0071】メモリー750、850及び950は、関連するプラットフォームのための一般的な記憶のようないくつかの機能を与える。他の機能は、各プロセッサ710、810及び910による実行の前に、アプリケーション720、820及び920、M1要素730、830及び930、並びにオペレーティングシステム740、840及び940を記憶することである。加えて、メモリー750、850及び950の一部は、ネットワーク50中の全てのプラットフォーム700、800及び900が使用可能な共有メモリーを構成してもよい。

【0072】(E、MIサービス)本発明は、クライアント/サーバーモデルを用いて実行してもよい。クライアントは、ダーティーコールやクリーンコールのような要求を発生させ、サーバーは、その要求に対して応答する。

【0073】図7に示す各MI要素730、830及び930は、クライアント要素とサーバー要素の双方を含んでいることが望ましい。図8は、クライアントプラットフォーム1000及びサーバープラットフォーム1100のブロック図を示しており、図7中のプラットフォーム700、800及び900のいずれか二つに当てはまる。

【0074】プラットフォーム1000及び1100は、それぞれ、メモリー1050及び1150、並びにプロセッサ1010及び1110を有している。プラットフォーム1000及び1100中の要素は、図7を参照しながら前述した同様の要素と同じ方法で動作する。この例では、プロセッサ1010は、クライアントアプリケーション1020を実行し、プロセッサ1110は、サーバーアプリケーション1120を実行する。プロセッサ1010及び1110は、また、それぞれ、オペレーティングシステム1040及び1140、並びにMI要素1030及び1130を実行する。

【0075】MI要素1030及び1130は、それぞれ、サーバーコールプロセッサ1031及び1131、アプリケーションコールプロセッサ1032及び1132、並びにガーベッジコレクター1033及び1133を有している。各MI要素1030及び1130は、また、各MI要素1030及び1130が監視する各リファレンスのための、リファレンスデータポーション1034及び1134、リファレンスカウント1035及び1135、並びに承認期間レジスタ1036及び1136を含むリファレンス要素を有している。

【0076】アプリケーションコールプロセッサ1032及び1132は、クライアントサービスに相当し、それぞれ、サーバーサービスに相当するサーバーコールプロセッサ1031及び1131と相互に情報伝達を行う。プラットフォーム1000及び1100は、サーバーコールプロセッサ、アプリケーションコールプロセッサ、ガーベッジコレクター及びリファレンス要素を有しているので、いずれのプラットフォームも、クライアントあるいはサーバーとして機能することができる。

【0077】しかしながら、後述する議論のために、プラットフォーム1000は、クライアントプラットフォームを示し、プラットフォーム1100は、サーバープラットフォームを示すものとする。この例では、クライアントアプリケーション1020は、分散リソースに対してリファレンスを取得し、サーバープラットフォーム1100のMI要素1130によって管理されるリソ

スに対してダーティーコールを送るために、MI要素1030を使用する。

【0078】加えて、サーバープラットフォーム1100は、サーバーアプリケーション1120を実行する。サーバーアプリケーション1120は、また、ダーティーコールを送信するためにMI要素1130を使用する。そして、ダーティーコールは、そのダーティーコールのリソースがMI要素1130によって管理されている時は、MI要素1130によって処理される。あるいは、サーバーアプリケーション1120は、MI要素1030で管理されているリソースに対してダーティーコールを送信するためにMI要素1130を使用する。

【0079】従って、クライアントプラットフォーム1000中のMI要素1030のためのサーバーコールプロセッサ1031、ガーベッジコレクター1033及びリファレンスカウント1035は、使用されていないので、図8中においては陰影が付されている。同様に、サーバープラットフォーム1100中のMI要素1130のアプリケーションコールプロセッサ1132もまた、休止状態であるので、陰影が付されている。

【0080】クライアントアプリケーション1020があるリソースに対応するリファレンスを取得した時には、アプリケーションコールプロセッサ1032は、ダーティーコールを送信し、それをサーバーコールプロセッサ1131が受信する。ダーティーコールには、要求されたリース期間が含まれている。サーバーコールプロセッサ1131は、ダーティーコール中のリファレンスのためにリファレンスカウント1135を増加させ、承認期間を決定する。これに応じて、サーバーコールプロセッサ1131は、アプリケーションコールプロセッサ1030に対して承認期間と共にリターンコールを送信する。アプリケーションコールプロセッサ1032は、記録されている承認期間1035を更新するため、及びダーティーコールのリファレンスに対応するリソースをいつリクレームするかを決定するためにその承認期間を使用する。

【0081】サーバーコールプロセッサ1131は、また、それが管理するリソースのためのリファレンスに対応するリファレンスカウントと承認期間を監視する。そのリファレンスカウント1135の内の一つがゼロの時、又はそのリファレンスのための承認期間1135が終了した時のいずれかの事象が最初に生じた場合は、いつでも、サーバーコールプロセッサ1131は、リファレンスカウントがゼロ又は承認期間が終了したリファレンスに対応するリソースをリクレームするためにガーベッジコレクター1133を起動してよい。

【0082】本発明の実施の形態に係るリース・リファレンス機構は、プロトコルに含まれるプラットフォーム1000及び1100上のクロックが同期していることを要求していない。その機構は、それらが比較しうる増

加期間を有することを、単に要求するだけである。リースは、特定の時刻に終了するのではなく、むしろ一定の時間間隔が経過した後に終了する。時間間隔がほぼ一致している限り、プラットフォーム1000及び1100は、承認されたリース期間について、およその合意が得られる。さらに、リースのためのタイミングは、コンピュータの見地からすれば、かなり長いので、クロックレートのわずかな差異は、ほとんど、あるいは全く影響がない。

【0083】ダーティーコールの転送時間は、プロトコルの影響を受ける。もし、MI要素1030がリファレンスに対するリースを保持し、更新の要求をするために、リースが終了する直前までウェイト状態であるならば、リースは、MI要素1130がその要求を受信する前に終了するかもしれない。もし、そうであるならば、MI要素1130は、更新要求を受信する前にリソースをリクレームするかもしれない。すなわち、ダーティーコールを送るときには、送り手側は、そのリソースのためのリース期間が終了する前に更新のダーティーコールがなされるように、ダーティーコールのリソースを処理しているプラットフォームへの転送時間を考慮して、要求されたリース期間に対して、時間因子を加えるべきである。

【0084】(F. 結論)本発明によれば、分散ガーベッジコレクション機構は、承認されたリース期間が終了した時に、リソースに対するリファレンスが終了するように、分散処理システム中のリソースに対するリファレンスに対応する承認されたリース期間を与えることによって、リファレンスの健全性を保証し、記憶漏れを排除する。その後に、そのリソースを収集するようにしてもよい。また、リソースに対するリファレンスに付与されたカウンターを参照して、リソースがもはや分散処理システム中のプロセスによってリファレンスされない時に、リソースを収集するようにしてもよい。

【0085】本発明は、上気した実施の形態に何ら限定されるものではなく、本発明の要旨を逸脱しない範囲で種々の改変が可能である。例えば、前述の実施の形態には、ソフトウェアが含まれているが、本発明は、ハードウェアとソフトウェアの組み合わせ、あるいはハードウェアのみで実施することができる。

【0086】

【発明の効果】本発明によれば、リファレンスの健全性は、ある一定の期間、すなわち、分散システム中のパー

ティ、例えば、リソースに対するリファレンスを保持しているアプリケーション及びそのリソースを管理しているガーベッジコレクションシステムが、そのリソースとそのリソースに対するリファレンスとが保証されることについて同意している期間、リソースをリースすることによって、大きな記憶漏れを伴うことなく保証される。

【0087】そのリース期間が終了したときには、そのリソースに対するリファレンスが継続するという保証は、消滅し、ガーベッジコレクションシステムにそのリソースをリクレームすることを許可する。なぜならば、そのリソースに対するリファレンスを保持しているアプリケーションと、そのリソースを管理しているガーベッジコレクションシステムは、有限の保証されたリース期間について合意しており、双方とも、いつそのリースが終了するか、すなわちいつその保証が終了するかを知ることができるからである。これにより、リファレンスのリース期間中はリファレンスの健全性が保証され、ネットワークエラーに起因して発生するリソースの開放失敗を回避することが可能となるという効果がある。

20 【図面の簡単な説明】

【図1】本発明の実施の形態に係る、アプリケーションコールプロセッサによって実行されるステップを示すフロー図である。

【図2】本発明の実施の形態に係る、ダーティーコールを処理するためのサーバーコールプロセッサによって実行されるステップを示すフロー図である。

【図3】本発明の実施の形態に係る、クリーンコールを処理するためのサーバーコールプロセッサによって実行されるステップを示すフロー図である。

30 【図4】本発明の実施の形態に係る、ガーベッジコレクションプロセスを起動するためのサーバーコールプロセッサによって実行されるステップを示すフロー図である。

【図5】分散処理システム内でのコールの好ましい流れを示す図である。

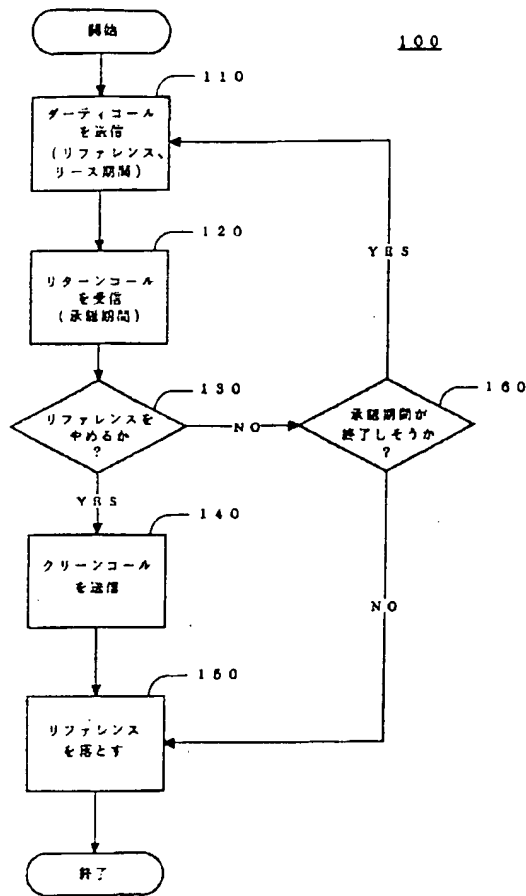
【図6】本発明に係る方法呼出作業を遂行する構成要素のブロック図である。

【図7】本発明の実施の際に用いられる分散処理システムを示す図である。

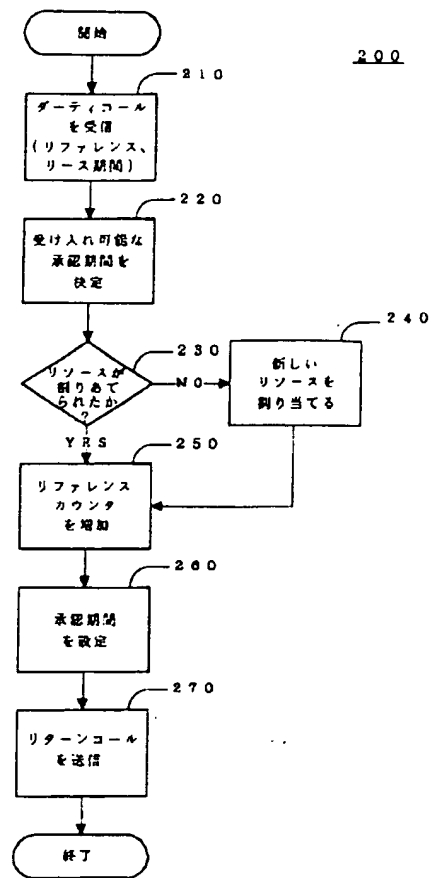
40 【図8】本発明の実施の形態に係る、分散処理システムのプラットフォームにおける個々のソフトウェアの構成要素を示す図である。



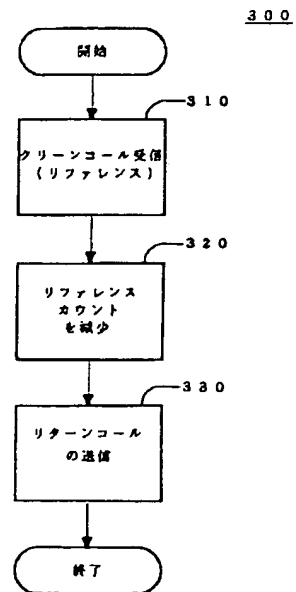
【図1】



【図2】

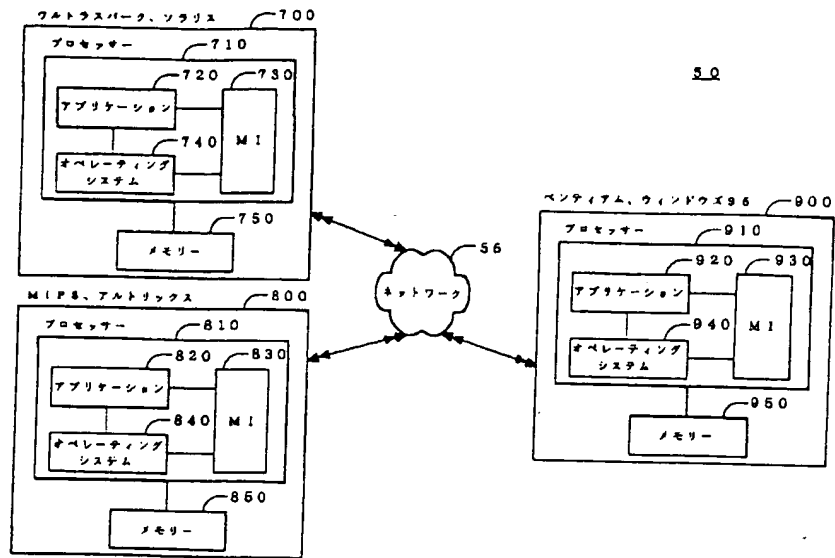


【図3】

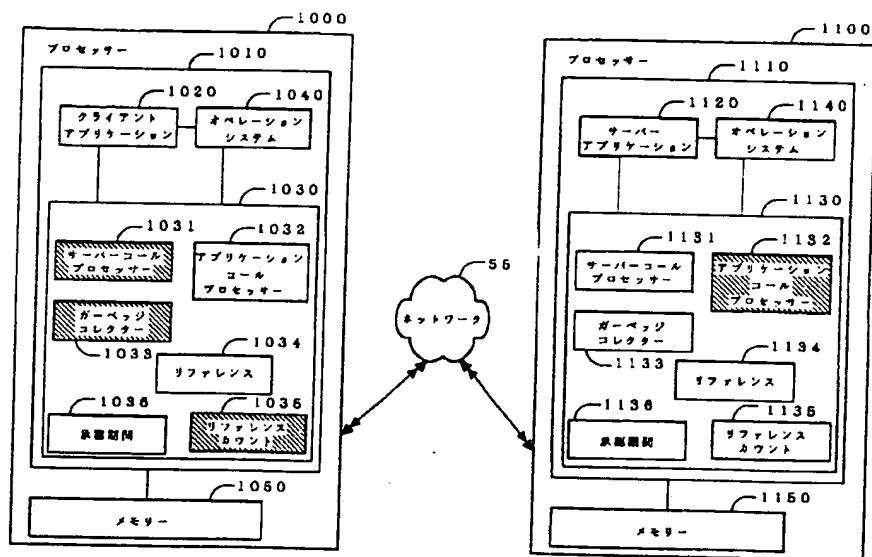




【図7】



【図8】



フロントページの続き

(72)発明者 ウォールドー ジェームス エイチ  
 アメリカ合衆国、01826 マサチューセッ  
 ツ州、ドラカット、ラビーロード155

(72)発明者 リッグス ロジャー  
 アメリカ合衆国、01826 マサチューセッ  
 ツ州、バーリントン、ブライアウッドレイ  
 ン4

- (54)【発明の名称】 分散ガーベッジコレクションのためのリソースの管理方法、装置、コンピュータプログラム生産物、コンピュータシステム並びにプラットフォーム間に分散したリソースの割当及び割当解除方法